

HIERARCHICAL PLAYLIST GENERATOR

This invention relates to the field of entertainment systems, and in particular to a playlist generator that distinguishes between general and specific preferences.

Playlists define a subset of entertainment selections, such as songs, videos, multimedia segments, and so on, for subsequent rendering via a corresponding rendering device or system.

A playlist generator facilitates the creation of a playlist. For ease of reference, the invention is presented in the context of a playlist generator that creates a playlist of songs. One of ordinary skill in the art will recognize that the principles presented herein are independent of the type of material being playlisted. Conventionally, a playlist generator receives a set of user preferences, and applies this set of preferences to a collection of material to identify selections in the collection that satisfy the set of preferences. The preferences may define songs that are to be included or excluded from the playlist, particular genres or styles, and so on, and may define a song to be used as a seed to find similar songs. To create a different playlist, a user modifies the preferences and submits this modified set of preferences to the playlist generator.

Generally, a selection process based on user preferences is not an "absolute" filtering process; often the user's preferences do not form a necessarily coherent or consistent set of criteria. For example, a user may indicate a preference for guitar music, ballads, and female vocalists. This does not necessarily imply that the user would not be interested in a guitar ballad performed by a male vocalist, or would not be interested in a ballad by a female vocalist without guitar backup. Most preference-based selection processes, therefore, use some form of weighted averaging to score each song relative to the given set of preferences, and songs that achieve a score above a given threshold value are identified as the songs that correspond to a user's preference. In the above example, a guitar ballad performed by a female vocalist would likely score highly, but other preferred characteristics may cause the guitar ballad performed by a male vocalist to score higher, and thus be more preferable to the user.

Alternative schemes, including rule-based systems, neural networks, and the like, are similarly based on providing a measure of correspondence between the characteristics

of each song and the given user preferences, and identifying those having the best measure of correspondence.

Hundreds of thousands of songs are currently available for legal downloading via the Internet, and the number is expected to continue to substantially increase. When a user initially submits a set of preferences to a song-searching playlist generator, the user typically receives a very large playlist, and must subsequently refine his/her preferences until the playlist size becomes manageable and/or usable. This iterative process is not always straightforward and/or productive, because the effects of a change of preference cannot easily be predicted, particularly in the context of a multi-parameter playlist generator. Often, the iterative changing of a variety of preference parameters results in a playlist that lacks any sense of unity or coherency. For example, a user may have previously identified a preference that effectively caused a particular group of songs to be excluded, but a new preference may affect the scoring scheme so as to reintroduce songs from this group of previously excluded songs back into the proposed playlist.

When the user next desires a different playlist, for example, due to a change in the user's mood or environment, the user either starts from scratch and progressively adds preference constraints until a desired playlist size is achieved, or, the user modifies the prior set of preferences in an attempt to maintain a reasonably sized playlist with some variety compared to the previous playlist. In either event, these playlist generation processes can be very time consuming, as each iterative playlist generation is achieved by applying the revised set of preferences to the hundreds of thousands of songs. Also, because each new set of preferences is applied to the entirety of the collection of songs, a modification that a user intends to further limit the selection process may, in fact, introduce additional selections into the playlist.

Automatic playlist generators based on vague preferences are intended to alleviate such difficulties. However, when the collection of material increases in size, the complexity of the task increases substantially, and the time to search and process the collection becomes excessive. To address the complexity issue, a class of playlist generators, commonly termed "recommenders", have been developed that provide playlists based on user's "tastes", rather than particular preferences, wherein a user's taste is defined as long-term characteristics, such as a general like for "rock and roll", "jazz",

and so on. However, playlists based on these general characteristics are often not coherent, and may not be consistent with the user's current mood or environment.

It is an object of this invention to provide a playlist generator that is scalable for efficiently generating playlists from ever-increasing collections of available material. It is a further object of this invention to provide a playlist generator that easily accommodates changes to a person's immediate preferences.

These objects and others are achieved by a playlist generator that partitions the playlist generation process into a multi-stage, hierarchical process. At an example highest hierarchical level, the user's "general" preferences are applied. These general preferences reflect the user's long-term tastes, and are relatively constant. At a lower level of the hierarchy, the user's "specific" preferences are applied. These specific preferences reflect the dynamic time-dependent or event-dependent aspects of a user's entertainment preferences. The high level preferences generate a subset of material from the global collection of material, and the lower level preferences are applied to this subset of material as the user's immediate preferences change. By segregating the playlist generation process hierarchically, efficiency is gained by applying immediate preference changes to a smaller subset of material, and coherency is improved by limiting the subsequent selection processes to a given subset of the material.

The drawings are included for illustrative purposes and are not intended to limit the scope of the invention. In the drawings:

FIG. 1 illustrates an example block diagram of a hierarchical playlist generator in accordance with this invention.

FIG. 2 illustrates an example block diagram of another hierarchical playlist generator in accordance with this invention.

Throughout the drawings, the same reference numerals indicate similar or corresponding features or functions.

The playlist generator 100 of FIG. 1 includes at least two selectors, 130, 150, that are configured to select material from one or more sources 110 to form a playlist 160 via a sequential filtering process. Optionally, the playlist generator 100 may also include a rendering device 190 for rendering the material identified in the playlist 160, or the playlist 160 may be provided to a separate rendering system. Although only two selectors

130, 150 are illustrated in FIG. 1, one of ordinary skill in the art will recognize that multiple selectors may be placed in a sequential path between sources 110 of content material and the generated playlist 160, in view of this disclosure.

A first selector 130, hereinafter termed a “recommender”, is configured to access the one or more sources 110 of content material, or indexes to the one or more sources 110, to select a subset of the available content material, based on a first set of preferences 170 of a user. In one embodiment of this invention, the first set of preferences 170 correspond to the general “tastes” of the user. These tastes are typically the time-invariant preferences of the user. For example, a particular user may generally like rock-and-roll, country, and classical music, and generally dislike jazz and folk music. That is, although at particular times, or during particular events, this user may prefer country over classical music, this user’s general tastes includes both country and classical music; but rarely, if ever, will a time or event arise when this user’s preference for music ever include jazz or folk music. In like manner, a user may generally like one particular artist, and absolutely dislike another artist, regardless of the genre of the song that these artists record. Similarly, a user may have a general preference or aversion to songs that include drum-solos.

To create the preferences, the user is provided a means for identifying his/her general likes and dislikes (not illustrated), typically via an interactive questionnaire. Although the above examples are presented in terms of absolute likes and dislikes, the questionnaire can be structured in the form of a weighting system, such as “on a scale of 0 to 10, please identify your preferences for each of the following”. Alternatively, or additionally, the tastes of a user may be deduced by observing the selections of content material by the user over an extended period of time, under a variety of circumstances.

The recommender 130 is configured to select content material from the one or more sources 110, or indexes to sources 110, that satisfy the general tastes of the user, to form a subset 140 of identifiers of the content material in the sources 110 that satisfy the user’s general tastes 170. Because the user’s general tastes include the user’s preferences regardless of a particular time or circumstance, the subset 140 can be expected to include a fairly substantial portion of the available material from the sources 110. For example, in a typical Internet-based environment wherein hundreds of thousands of songs are

available for inclusion in the playlist 160, the subset 140 may include thousands or tens of thousands of songs that satisfy a particular user's general tastes 170.

Of particular note, due to the sequential nature of the selectors 130, 150, material that does not satisfy the user's general tastes 170 are not included in the subset 140, and thus are not available for selection by the selector 150, regardless of subsequent user preferences 180, discussed below. This high-level or coarse filtering of the available material from the sources 110, provides for efficient processing by the subsequent selector 150, and also eliminates the possibility of a subsequent selection including material that does not conform to the user's general tastes.

The subsequent selector 150, or series of subsequent selectors, selects a further subset of identifiers of material from the subset 140, based on a second set of user preferences 180. These user preferences 180 are generally time-dependent, or event-dependent, or merely based on a particular mood of the user at the time the playlist 160 is generated. For example, a user who generally likes rock-and-roll may prefer not to hear rock-and-roll in the early morning hours. In like manner, a user who likes rock-and-roll and country music may further refine the selection to include only romantic songs for a playlist intended for use during a romantic event. Similarly, particular music characteristics may be preferred for playlists intended for the dinner hour, and other characteristics for playlists intended for driving in rush-hour traffic.

As noted above, because the second selector 150 applies the set of preferences 180 to the subset 140 that conforms to the user's general tastes, the risk of including unwanted material in the playlist 160 is considerably reduced compared to a conventional single-stage playlist generator. For example, in a conventional system a user may specify that he/she does not prefer jazz, but because the filtering is based on a composite of preferences, this no-jazz preference may be overcome by an expressed preference for items such as "instrumentals", "strong beat", "horns", and so on, for a particular occasion or particular mood of the user. In the embodiment of FIG. 1, the no-jazz preference will typically be applied at the higher-level selector 130, and the event-specific or mood-specific preference for the particular set of music characteristics will be included in the specific time-dependent or event-dependent preferences 180. As such, because the subset 140 is formed based on the general "no-jazz" preference, the subset 140 will not contain

jazz selections. Thereafter, regardless of the inclusion of particular music characteristics in the specific preferences 180, the playlist 160 cannot contain jazz selections.

Although the invention has been presented using the paradigm of a recommender module 130 that provides a first subset of selections based on a user's general preferences, one of ordinary skill in the art will recognize that other schemes may be used. For example, FIG. 2 illustrates a hierarchical playlist generator 200 that includes a recommender module 230 that is configured to search for songs that are currently popular, wherein popularity is defined in terms of the number of times other users have accessed each song during a given time period.

In FIG. 2, a song-finder 215 is illustrated to represent one or more web-based applications that multiple users 225 use to access available content material in the sources 110. The popularity logger module 230 in the user's system 200 accesses the song-finder 215 to determine the popularity of songs, based on the number of accesses to each item in the sources 110 via the source-finder 215 within a given time period.

In this embodiment, the user's preferences 270 include parameters for filtering the sources 110 for popular songs, and may include, for example, a time duration (previous day, previous week, previous month, etc.) for determining the tally of the number of requests for each song, a minimum number of accesses to qualify as a 'popular' song, and so on. The user is also provided the option of specifying the extent of the list, in terms of the number of songs that are to be included in the popular sublist 140. Typically, in a system wherein the sources 110 contain hundreds of thousands of songs, a user will configure the logger 230 to identify a few thousand popular songs, so that a sufficient variety is obtained for subsequent filtering based on the user's specific preferences, discussed further below.

Optionally, depending upon the extent of the records maintained at the song-finder 215, the logger module 230 may be configured to determine the popularity of songs among a given subset of the users of the song-finder 215. In such an embodiment, the user of the system 200 identifies general tastes, such as "rock-and-roll", "jazz", "classical", and so on, as discussed above, and the popularity logger module 230 determines the songs that were most requested by other users who have tastes in common with the user's tastes. Similarly, the user preferences 270 can include the user's general

tastes 170, so that the logger module 230 includes only songs that fit the user's general tastes 170 in the popular sublist 140. That is, for example, if the user's taste includes country-swing and rock-and-roll, the logger 230 can optionally be configured to report the N most-popular songs that are classified as either country-swing or rock-and-roll, wherein most-popular is determined by the number of times the song has been requested by a user in a given time period. Similarly, the user can identify classifications of songs that should be specifically excluded from consideration in forming the popular list 140, such as not-Jazz, or not-Madonna, etc.

Note that by structuring the popular list 140 based on the user's general tastes, the popular list 140 is not dependent upon a predetermination of which material to classify as 'popular', as in conventional internet-radio and other playlist generators. As such, the popular list 140 can be as narrowly-defined or broadly-defined as the user prefers. For example, a user may request the most popular hard-rock songs, the most popular piano concertos, the most popular Italian folk songs, the most popular children songs, and so on. In like manner, by structuring the popular list 140 based on requests by other users, the popular list 140 will likely automatically include seasonal-appropriate songs, new songs from particular artists, and so on.

The selector 150 is configured to generate the playlist 160 from the popular list 140, based on the user's specific preferences 180 for the particular playlist, as detailed above. If the user's general tastes 170 are not reflected in the songs selected by the popularity logger 230, these general preferences may be included in the preferences 180, or these general preferences may be included in a mid-level of the hierarchical scheme illustrated in FIG. 2, to generate an intermediate list, after the popular list 140, and the selector 150 will be configured to select the songs for the playlist 160 from this intermediate list, based on the user's specific preferences 180.

In a typical embodiment of this invention, the subset 140 is stored in non-volatile storage, so that multiple playlists 160 can be generated from this subset 140 as the user desires. For example, the system 100 may be configured to apply the selector 130/230 to the available sources 110 on a weekly basis, to keep the subset 140 relatively up to date. During the week, the user applies the selector 150 to the subset 140 to generate playlists 160 as required. Because the subset 140 is a collection of identifiers that is substantially

smaller than a collection of identifiers of all of the material at the sources 110, and because the subset 140 is typically generated at the user's system, the repeated operations of the selector 150 is substantially more efficient than the conventional repeated operation of a single-stage selector that searches the entire collection at the sources 110.

As noted above, the selector 150 may include multiple sequential selectors. For example, a recommender based on general tastes may follow the aforementioned popular-song recommender 230 to provide a multi-tiered filter process before the selector 150 is applied to select songs based on current preferences 180. In like manner, a selector may be configured to create a sub-subset from the subset 140 based on the time-of-day, and then another selector is configured to create the playlist 160 from this sub-subset based on the user's current mood during this time-of-day. In this manner, the user need not repeatedly enter the time-of-day preferences for each change of current mood. These and other techniques for organizing preferences and selectors in a hierarchical manner will be evident to one of ordinary skill in the art in view of this disclosure.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements which, although not explicitly described or shown herein, embody the principles of the invention and are thus within its spirit and scope. For example, a user may create a collection of preference files, such as "general", "romantic", "lively", "dinner", "blues", and so on, and may selectively apply these preferences in a sequential manner. By applying the preferences in a sequential manner and thereby generating progressively smaller subsets, the user is able to generate a progressively more selective playlist without the risk of a particular preference introducing a divergent search result. In like manner, although the invention is presented herein with the "general" preferences being applied first, at the highest level, and the "specific" preferences applied subsequently, the system can be configured to allow the user to select the particular order of preference selection. These and other system configuration and optimization features will be evident to one of ordinary skill in the art in view of this disclosure, and are included within the scope of the following claims.